MPAC-100
RS-232 PCI
SYNCHRONOUS ADAPTER

for PCI Card Standard compatible machines

User's Manual

QUATECH, INC.
662 Wolf Ledges Parkway
Akron, Ohio 44311

TEL: (330) 434-3154
FAX: (330) 434-1409
www.quatech.com

WARRANTY INFORMATION

Quatech Inc. warrants the **MPAC-100** to be free of defects for **one (1) year** from the date of purchase. Quatech Inc. will repair or replace any board that fails to perform under normal operating conditions and in accordance with the procedures outlined in this document during the warranty period. Any damage that results from improper installation, operation, or general misuse voids all warranty rights.

Although every attempt has been made to guarantee the accuracy of this manual, Quatech Inc. assumes no liability for damages resulting from errors in this document. Quatech Inc. reserves the right to edit or append to this document at any time without notice.

Please complete the following information and retain for your records. Have this information available when requesting warranty service.

DATE OF PURCHASE:

MODEL NUMBER: **MPAC-100**

PRODUCT DESCRIPTION: **Single Channel PCI RS-232-D Synchronous Communications Adapter**

SERIAL NUMBER:

Copyright 2000 Quatech, Inc.

NOTICE

The information contained in this document is protected by copyright, and cannot be reproduced in any form without the written consent of Quatech, Inc. Likewise, any software programs that might accompany this document are protected by copyright and can be used only in accordance with any license agreement(s) between the purchaser and Quatech, Inc. Quatech, Inc. reserves the right to change this documentation or the product to which it refers at any time and without notice.

The authors have taken due care in the preparation of this document and every attempt has been made to ensure its accuracy and completeness. In no event will Quatech, Inc. be liable for damages of any kind, incidental or consequential, in regard to or arising out of the performance or form of the materials presented in this document or any software programs that might accompany this document.

Quatech, Inc. encourages feedback about this document. Please send any written comments to the Technical Support department at the address listed on the cover page of this document.

Table of Contents

1 Introduction 7
1.1 System Requirements 7
2 Hardware Installation 8
3 Windows 95/98 Installation 9
4 Other Operating Systems 14
5 Using the MPAC-100 with Syncdrive 18
6 Addressing 19
7 Interrupts 20
8 SCC General Information 21
8.1 Accessing the registers 22
8.2 Baud Rate Generator Programming 25
8.3 SCC Data Encoding Methods 25
8.4 Support for SCC Channel B 26
8.4.1 Receive data and clock signals 26
8.4.2 Extra clock support for channel A 26
8.4.3 Extra handshaking for channel A 26
8.4.4 Other signals are not used 26
8.5 SCC Incompatibility Warnings 27
8.5.1 Register Pointer Bits 27
8.5.2 Software Interrupt Acknowledge 27
9 FIFO Operation 28
9.1 Enabling and disabling the FIFOs 28
9.2 Accessing the FIFOs 28
9.2.1 Transmit FIFO 28
9.2.2 Receive FIFO 29
9.3 SCC configuration for FIFO operation 29
9.3.1 Using channel A for both transmit and 30
9.3.2 Using channel B for receive 31
9.4 FIFO status and control 32
9.4.1 Interrupt status 32
9.4.2 Resetting the FIFOs 33
9.4.3 Reading current FIFO status 33
9.4.4 Controlling the FIFOs 33
9.5 Accessing the SCC while FIFOs are enabled 33
9.6 Receive pattern detection 34
9.7 Receive FIFO timeout 35
10 Communications Register 36
11 Configuration Register 38
12 Interrupt Status Register 40
13 FIFO Status Register 41
14 FIFO Control Register 42

Table of Contents

15 Receive Pattern Character Register 43
16 Receive Pattern Count Register 44
17 Receive FIFO Timeout Register 45
18 External Connections 46
18.1 5V fuse (pin 9) 47
18.2 SYNCA (pin 10) 47
18.3 RING (pin 22) 47
18.4 Null-modem cables 49
19 DTE Interface Signals 50
20 PCI Resource Map 53
21 Specifications 54

1 Introduction

The Quatech MPAC-100 is a PCI Type card and is PCI PC Card Standard Specification compliant. It provides a single-channel RS-232 synchronous communication port. The base address and IRQ are configured through the PCI hardware and software using utility programs provided by Quatech. There are no switches or jumpers to set.

The MPAC-100 uses a Zilog 85230-compatible Serial Communications Controller (SCC). The SCC can support asynchronous formats, byte-oriented synchronous protocols such as IBM Bisync, and bit-oriented synchronous protocols such as HDLC and SDLC. The SCC also offers internal functions such as on-chip baud rate generators, and digital phase-lock loop (DPLL) for recovering data clocking from received data streams.

Because the PCI standard does not include a direct memory access (DMA) interface, the MPAC-100 supports only interrupt-driven communications. To compensate for the lack of DMA, the MPAC-100 is equipped with 1024-byte FIFOs for transmit and receive data. The FIFOs provide for high data throughput with very low interrupt overhead.

1.1 System Requirements

- 16 bytes of contiguous I/O address space
- one hardware interrupt (IRQ)
- One available PCI expansion slot

2 Hardware Installation

Hardware installation for the MPAC-100 is a very simple process:

1. Turn off the power of the computer system in which the MPAC-100 is to be installed.
2. Remove the system cover according to the instructions provided by the computer manufacturer.
3. Install the MPAC-100 in any empty PCI expansion slot. The board should be secured by installing the Option Retaining Bracket (ORB) screw.
4. Replace the system cover according to the instructions provided by the computer manufacturer.
5. Attach and secure the cable connectors to the desired equipment.
6. Turn on the power of the computer system.

3 Windows 95/98 Installation

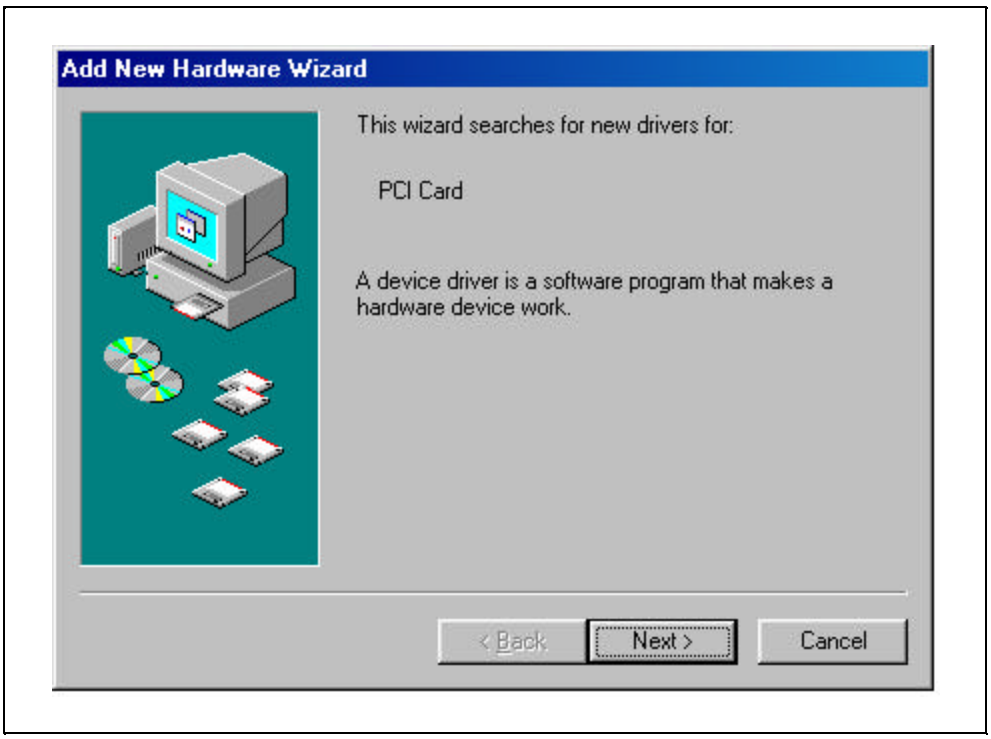
Windows 95/98 maintains a registry of all known hardware installed in your computer. Inside this hardware registry Windows keeps track of all of your system resources, such as I/O locations, IRQ levels, and DMA channels. The "Add New Hardware Wizard" utility was designed to add new hardware and update this registry.

An "INF" configuration file is included with the MPAC-100 to allow easy configuration in the Windows 95/98 environment. Windows uses the "INF" file to determine the system resources required by the MPAC-100, searches for available resources to fill the boards requirements, and then updates the hardware registry with an entry that allocates these resources. The Syncdrive DLL and VxD can then be used to access the card.

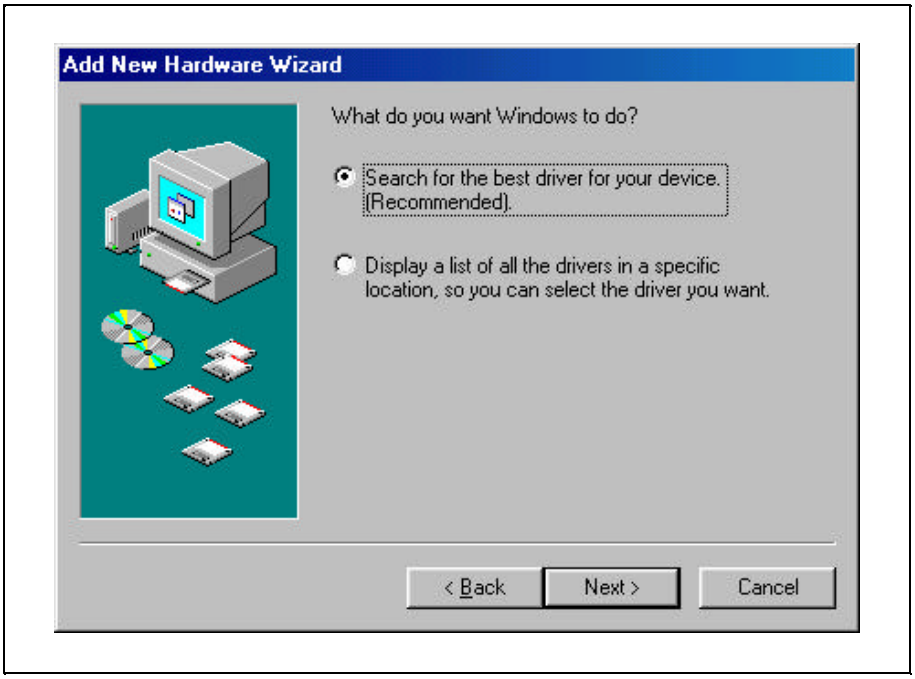
3.1 Using the "Add New Hardware" Wizard

The following instructions provide step-by-step instructions on installing the MPAC-100 in Windows 98 using the "Add New Hardware" wizard. Windows 95 uses a similar process to load the INF file from a CD with slightly different dialog boxes.

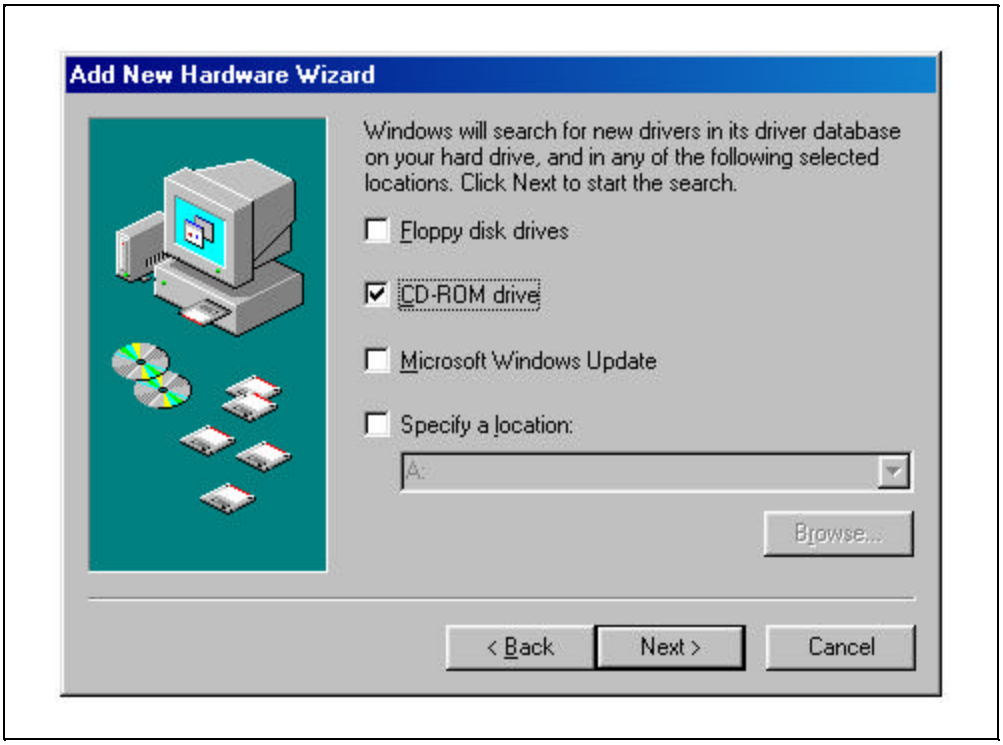
1. After inserting an MPAC-100 for the first time, the "Add New Hardware" wizard will start. Click the "Next" button.


Add New Hardware Wizard
This wizard searches for new drivers for:
PCI Card
A device driver is a software program that makes a hardware device work.
< Back
Next >
Cancel

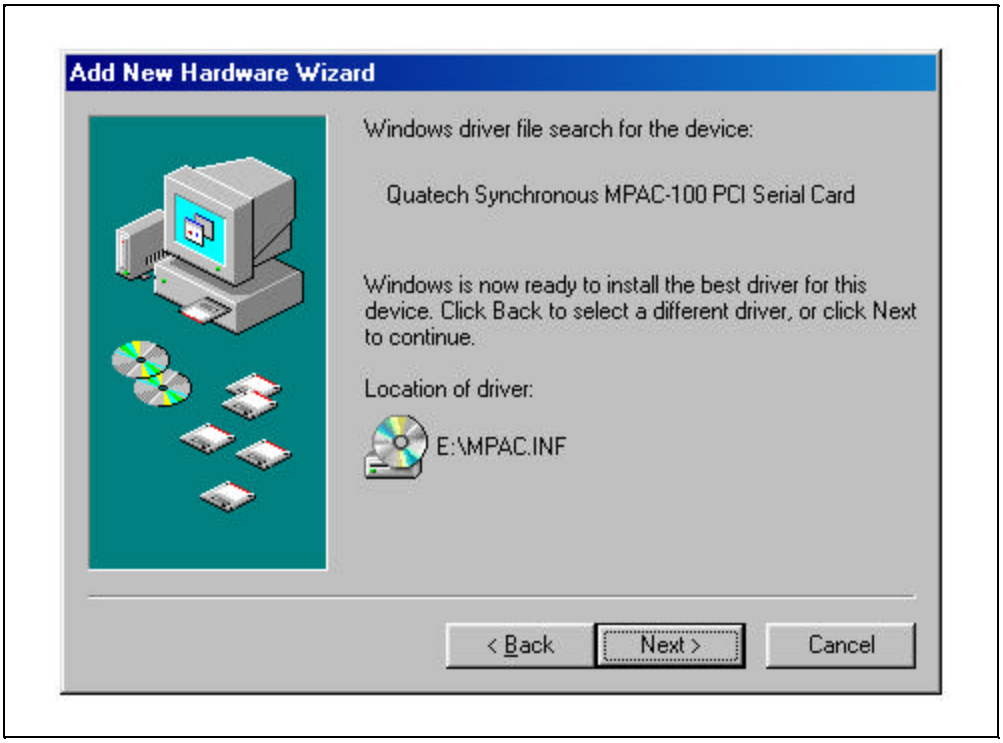
2. Click the "Next" button. Select the radio button for "Search for the best driver for your device." Click the "Next" button to continue.


Add New Hardware Wizard
What do you want Windows to do?
Search for the best driver for your device. (Recommended).
Display a list of all the drivers in a specific location, so you can select the driver you want.
< Back
Next >
Cancel

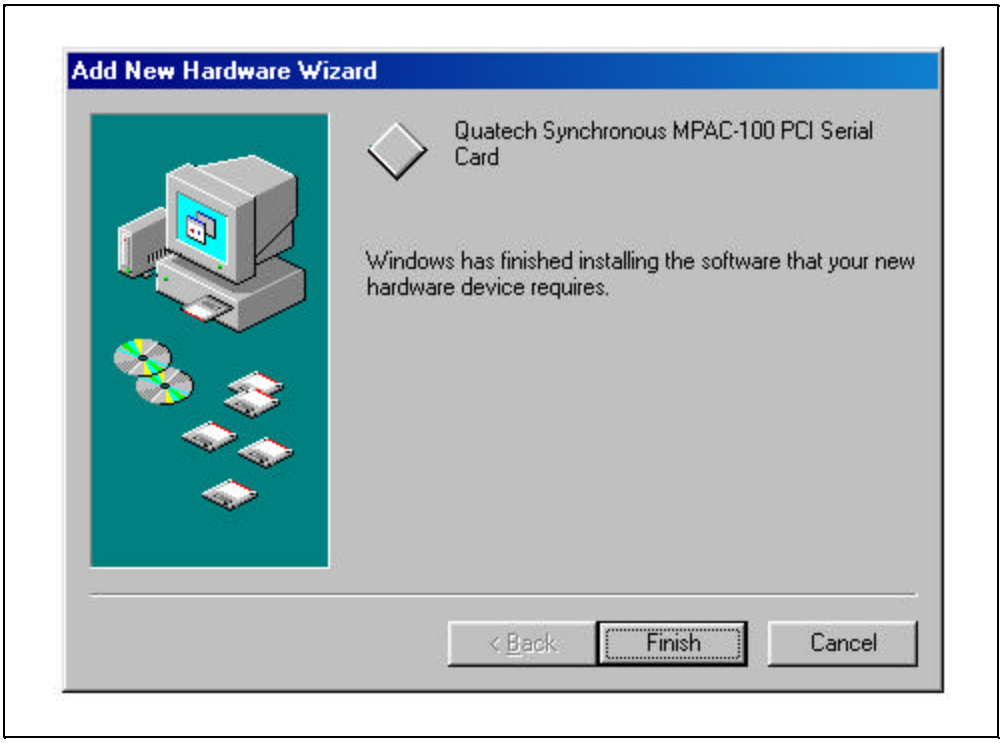
3. On the next dialog, select the "CD-ROM drive" checkbox. Insert the Quatech COM CD (shipped with the card) into the CD-ROM drive. Click the "Next" button.


Add New Hardware Wizard
Windows will search for new drivers in its driver database on your hard drive, and in any of the following selected locations. Click Next to start the search.
Floppy disk drives
CD-ROM drive
Microsoft Windows Update
Specify a location:
A:
Browse...
< Back
Next >
Cancel

4. Windows should locate the INF file on the CD and display a dialog that looks like this. Click the "Next" button.


Add New Hardware Wizard
Windows driver file search for the device:
Quatech Synchronous MPAC-100 PCI Serial Card
Windows is now ready to install the best driver for this device. Click Back to select a different driver, or click Next to continue.
Location of driver:
E:\MPAC.INF
< Back
Next >
Cancel

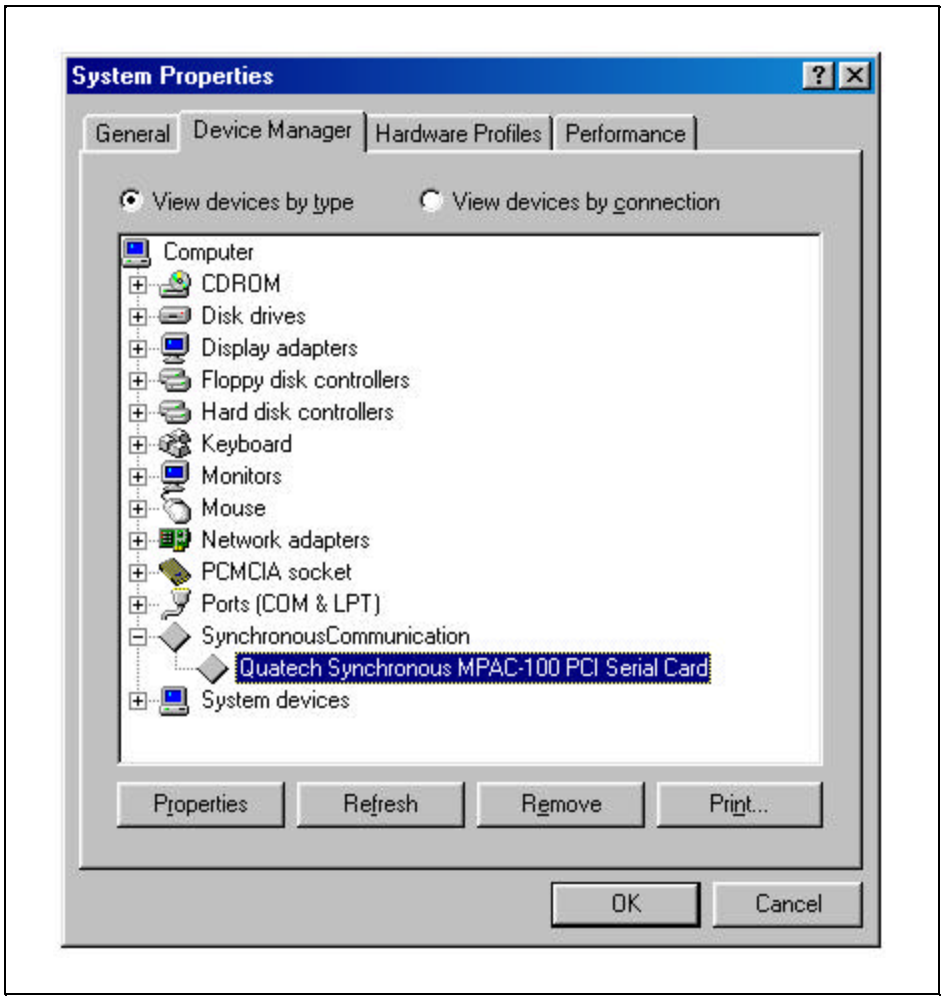
5. Windows will copy the INF file from the CD and display a final dialog indicating that the process is complete. Click the "Finish" button.


Add New Hardware Wizard
Quatech Synchronous MPAC-100 PCI Serial Card
Windows has finished installing the software that your new hardware device requires.
< Back
Finish
Cancel

3.2 Viewing Resources with Device Manager

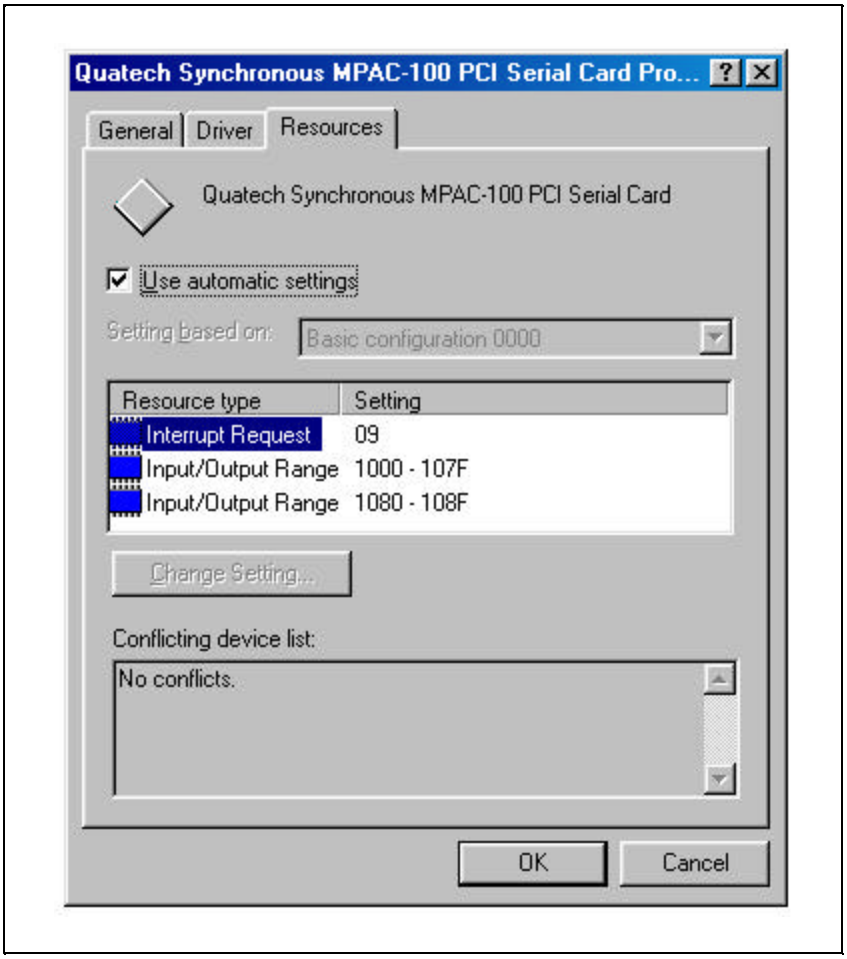
The following instructions provide step-by-step instructions on viewing resources used by the MPAC-100 in Windows 95/98 using the "Device Manager" utility.

1. Double click the "System" icon inside the Control Panel folder. This opens up the System Properties box.
2. Click the "Device Manager" tab located along the top of the System Properties box.
3. Double click the device group "Synchronous_Communication". The MPAC-100 model name should appear in the list of adapters.



4. Double click the MPAC-100 model name and a properties box should open for the hardware adapter.

5. Click the "Resources" tab located along the top of the properties box to view the resources Windows has allocated for the MPAC-100 match the hardware configuration. Click "Cancel" to exit without making changes.



6. If changes to the automatic configuration are necessary for compatibility with existing programs, uncheck the "Use Automatic Settings" box and double-click on the Resource Type that needs to be changed. Caution should be used to avoid creating device conflicts with other hardware in the system.

4 Other Operating Systems

Device drivers for Windows NT and OS/2 are also available for the MPAC-100. The board can be used under DOS and other operating systems as well in many circumstances. The software described below can be downloaded from the Quatech web site if it did not come with the board.

4.1 Windows NT

The Windows NT device driver is installed by running the SETUP program. Up to 256 serial ports are supported. There is a command line-based configuration utility which is used for adding PCI bus and ISA bus serial ports. Please refer to the documentation included with the device driver for full installation and configuration details.

4.2 OS/2

The OS/2 device driver supports up to 32 serial ports in a system. Installation is a manual, but simple, process. Please refer to the documentation included with the device driver for full installation and configuration details.

4.3 DOS and other operating systems

The MPAC-100 is not a direct drop-in replacement for a legacy serial port because its base address and IRQ cannot be fixed at values such as 3F8 hex, IRQ 4 (COM1) or 2F8 hex, IRQ 3 (COM2), etc. Rather, the system BIOS assigns the address and the IRQ in a plug-and-play fashion at boot time. Software which is to use the MPAC-100 must be able to accommodate any valid assignments of these resources.

For Windows 95, Windows NT and OS/2, the Quatech device drivers determine what the resource assignments are and proceed accordingly. In other cases, however, the user must intervene. The discussion below will center on DOS, but the concepts can be applied to other operating systems as well.

Many DOS applications support user configuration of the base address and IRQ of a serial port. Such applications can generally make use of the MPAC-100. Older applications, as well as some custom software, may use hard-coded standard legacy serial port addresses. These applications will require modifications if they are to use the MPAC-100.

Custom applications for which the customer has source code can be modified to make just a few PCI BIOS function calls to obtain all the necessary configuration information. The PCI BIOS specification can be
obtained from the PCI Special Interest Group. Contact Quatech technical support for more information.

4.4 QTPCI.EXE

Quatech's "QTPCI" utility supplies the information required when modifying the serial port settings of the application. This program should be run from real DOS, not in a Windows DOS box.

Figure 13 shows the Basic Mode display for the MPAC-100 after the "Q" key has been pressed. In this example, the MPAC-100 uses I/O base address FF80 hex and IRQ 11. The hardware revision of the MPAC-100 is also displayed. Pressing the "N" key will show similar information for all non-Quatech PCI devices in the system, including those devices integrated on the motherboard.

The QTPCI program is capable only of displaying the PCI configuration. It cannot be used to make changes.

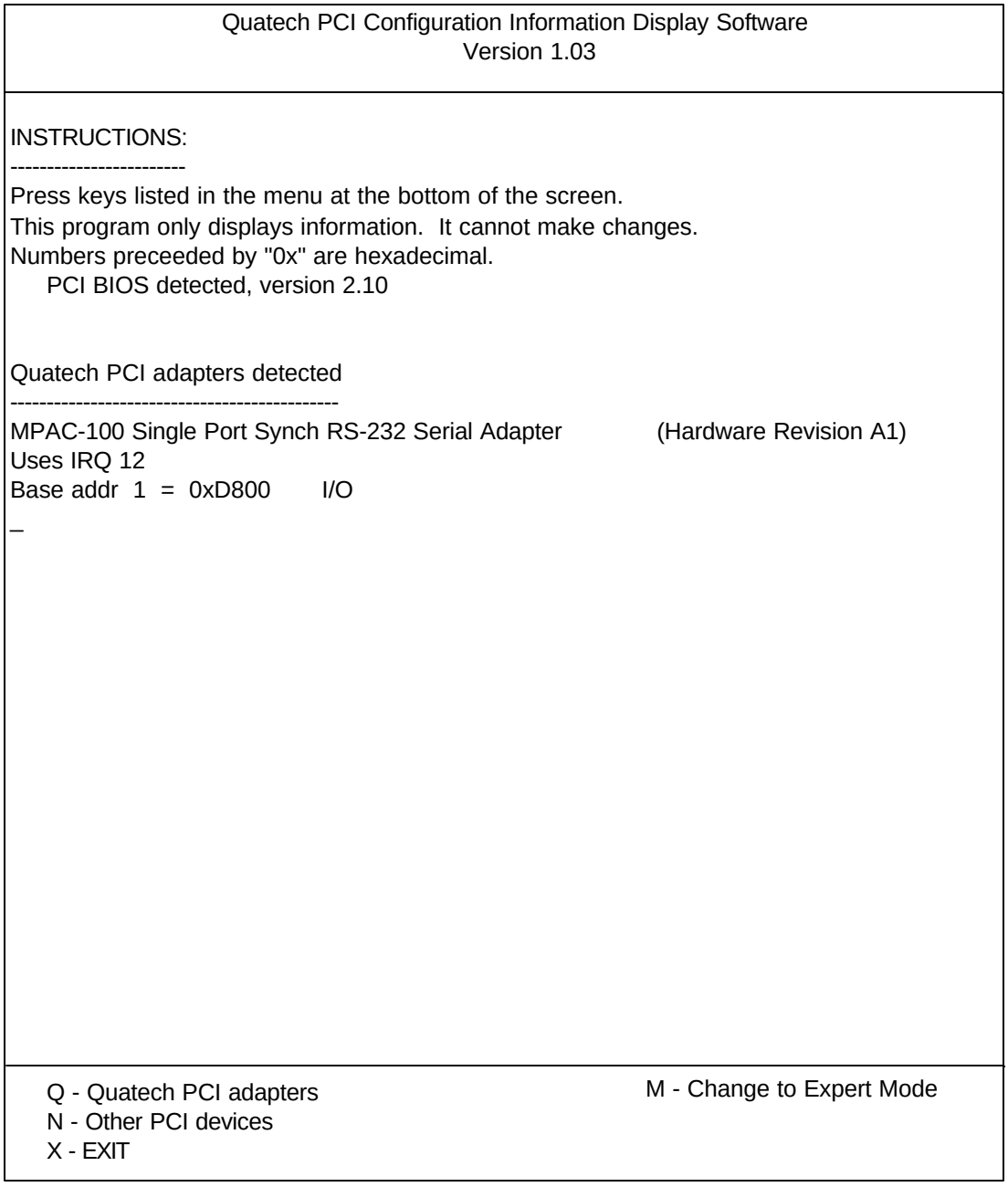

Quatech PCI Configuration Information Display Software
Version 1.03
INSTRUCTIONS:
Press keys listed in the menu at the bottom of the screen.
This program only displays information. It cannot make changes.
Numbers preceeded by "0x" are hexadecimal.
PCI BIOS detected, version 2.10
Quatech PCI adapters detected
MPAC-100 Single Port Synch RS-232 Serial Adapter (Hardware Revision A1)
Uses IRQ 12
Base addr 1 = 0xD800 I/O
Q - Quatech PCI adapters
N - Other PCI devices
X - EXIT
M - Change to Expert Mode

Figure 14 --- QTPCI.EXE Basic Mode display

Figure 14 shows the Expert Mode display for the MPAC-100 after the "Q" key has been pressed. The information from the Basic Mode display is presented along with more details such as the Vendor and Device IDs, PCI Class Code, size of memory

and I/O regions, etc. Pressing the "N" key will show similar information for all non-Quatech PCI devices in the system, including those devices integrated on the motherboard. In this example, the "Base addr 0" resource is reserved.

For users interested in even more details, PCI BIOS information can be displayed by pressing the "B" key. Pressing the "I" key displays the PCI interrupt routing table.

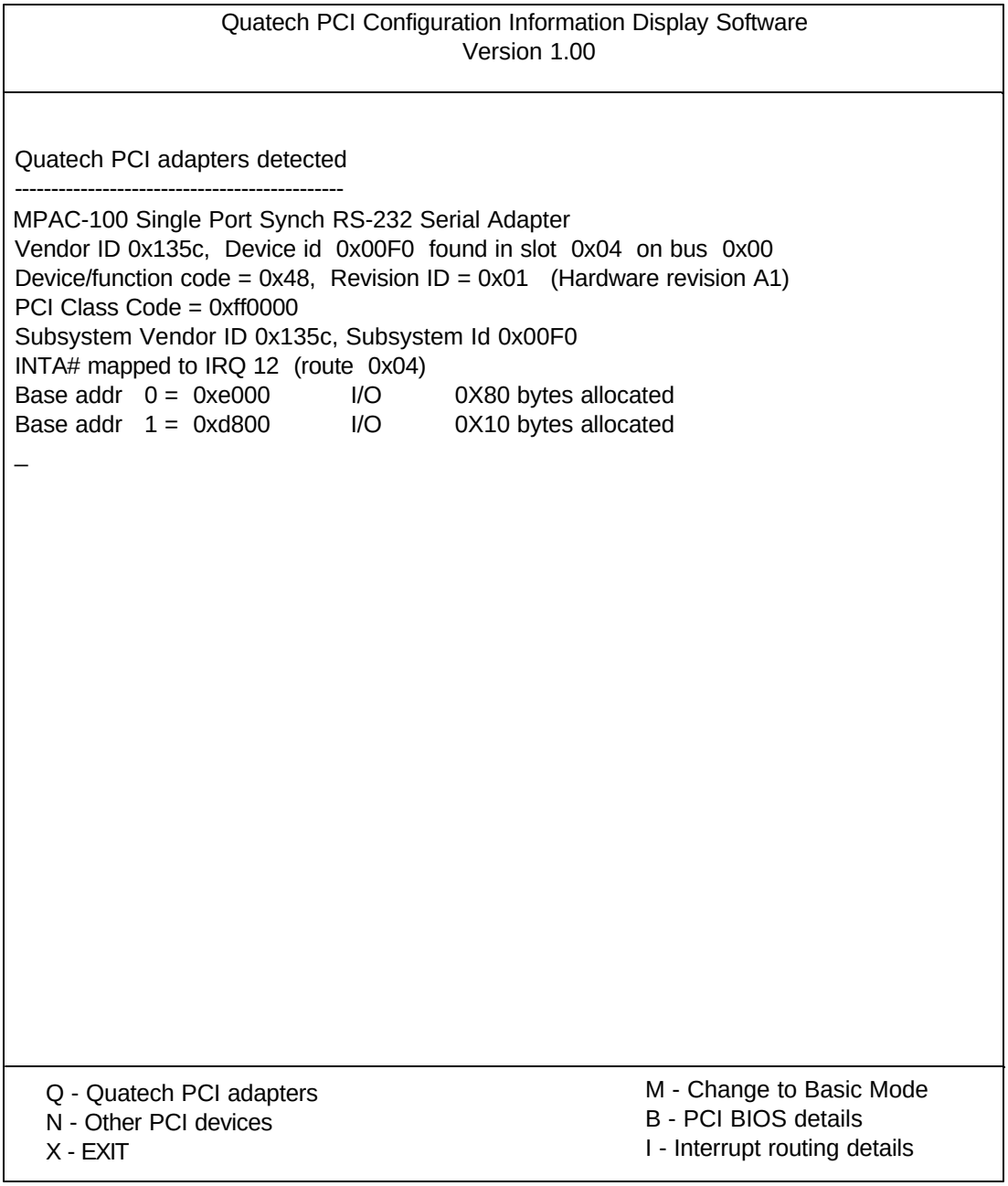

Quatech PCI Configuration Information Display Software
Version 1.00
Quatech PCI adapters detected
MPAC-100 Single Port Synch RS-232 Serial Adapter
Vendor ID 0x135c, Device id 0x00F0 found in slot 0x04 on bus 0x00
Device/function code = 0x48, Revision ID = 0x01 (Hardware revision A1)
PCI Class Code = 0xff0000
Subsystem Vendor ID 0x135c, Subsystem Id 0x00F0
INTA# mapped to IRQ 12 (route 0x04)
Base addr 0 = 0xe000 I/O 0X80 bytes allocated
Base addr 1 = 0xd800 I/O 0X10 bytes allocated
Q - Quatech PCI adapters
N - Other PCI devices
X - EXIT
M - Change to Basic Mode
B - PCI BIOS details
I - Interrupt routing details

Figure 15 --- QTPCI.EXE Expert Mode display

5 Using the MPAC-100 with Syncdrive

Syncdrive is a synchronous communications software driver package designed to aid users of Quatech synchronous communication hardware in the development of their application software. Syncdrive is included free of charge with all Quatech MPA-series synchronous communication products. The MPAC-100 is backward-compatible with software written for Quatech ISA-bus synchronous adapters and it operates with Syncdrive.

Syncdrive, however, is not aware of the plug-and-play nature of PCI cards. A Syncdrive application will expect to see the MPAC-100 at a specific base address and a specific IRQ. When using Syncdrive with PCI cards, it is necessary to obtain the base address and IRQ assigned to the card by using the QTPCI.EXE software utility previously discussed.

For DOS, Windows 3.1, or OS/2, QTPCI.EXE must be used to obtain the base address and IRQ needed by the Syncdrive application before the application tries to use the card.

Under Windows 95/98, the card is automatically configured. To find the settings, click the right mouse button on the My Computer icon and select Properties. Select the Device Manager tab and double-click the card's entry under the "Synchronous Communication" section. Select the Resources tab to see the card's base address and IRQ. Use these settings with the Syncdrive application. Windows 95/98 may allow changes to the settings if the "Use Automatic Settings" box is unchecked.

Syncdrive does not receive notifications of card insertion or card removal events. Therefore it cannot support hot swapping without the user taking some kind of action to force the Syncdrive application to initialize a newly-inserted card.

A future release of Syncdrive may permit automatic configuration by retrieving hardware settings from the MPAC-100. For now, the user should consider the QTPCI.EXE program, it reflects the location (base address and IRQ) the card is in before the Syncdrive application is started.

6 Addressing

The MPAC-100 occupies a continuous 16-byte block of I/O addresses. For example, if the base address is set to 300 hex, then the MPAC-100 will occupy address locations 300 hex to 30F hex. If the computer in which the MPAC-100 is installed is running PCI Card and Socket Services, the base address is set by the client driver. If PCI Card and Socket Services are not being used, the base address is set by the MPAC-100 enabler program.

The first four bytes of address space on the MPAC-100 contain the internal registers of the SCC. Other Quatech architecture-specific registers occupy eight more bytes. The remainder of the address space is reserved for future use. The MPAC-100 address map is shown in Table 2.

Address	Register Description
Base + 0	SCC Data Port, Channel A
Base + 1	SCC Control Port, Channel A
Base + 2	SCC Data Port, Channel B
Base + 3	SCC Control Port, Channel B
Base + 4	Communications Register
Base + 5	Configuration Register
Base + 6	Reserved
Base + 7	Reserved
Base + 8	Interrupt Status Register
Base + 9	FIFO Status Register
Base + A	FIFO Control Register
Base + B	Receive Pattern Character Register
Base + C	Receive Pattern Count Register
Base + D	Receive FIFO Timeout Register
Base + E	Reserved
Base + F	Reserved

Table 2 --- MPAC-100 Address Assignments

Information on the internal registers of the SCC can be found in Table 3 and Table 4 and in the technical reference manuals available from Zilog. The other onboard registers are fully described in subsequent chapters of this manual.

7 Interrupts

The MPAC-100 will operate using the interrupt level (IRQ) assigned by the PCI system. Interrupts can come from the SCC, the internal FIFOs or RS-232 test mode. The interrupt source is selected by bits 4 and 5 of the Configuration Register (see page 41).

When using interrupts with the MPAC-100, the application must have an interrupt service routine (ISR). There are several things that an ISR must do to allow proper system operation:

1. If the internal FIFOs are enabled, read the Interrupt Status Register (see page 43) to determine whether the interrupt was caused by a FIFO event or by the SCC.
2. If the TX_FIFO bit is set, at least 512 bytes can be written to the Tx FIFO. If the RX_FIFO bit is set, at least 512 bytes can be read from the Rx FIFO. I/O block move instructions may be useful. Check the FIFO Status Register (see page 44) after servicing the FIFO(s) to see if further FIFO service is required.
3. If the SCC bit is set, do an SCC software interrupt acknowledge by reading Read Register 2 in channel B of the SCC. The value read can also be used to vector to the appropriate part of the ISR.
4. Service the SCC interrupt by reading the receiver buffer, writing to the transmit buffer, issuing commands to the SCC, etc.
5. Write a Reset Highest Interrupt Under Service (IUS) command to the SCC by writing 0x38 to Write Register 0.
6. Check for other interrupts pending in the SCC by reading Read Register 3. Perform further interrupt servicing if necessary.
7. For applications running under DOS, a nonspecific End of Interrupt must be submitted to the interrupt controller. For Interrupts 2-7 this is done by writing a 0x20 to port 0x20. For Interrupts 10-12, 14 and 15 this is done by writing a 0x20 to port 0x60, then a 0x20 to port 0x20 (due to the interrupt controllers being cascaded). Device drivers running under other operating systems may have varying requirements concerning the End of Interrupt command.

For further information on these subjects or any others involving the SCC contact Zilog for a complete technical manual.

8 SCC General Information

The Serial Communications Controller (SCC) is a dual channel, multi-protocol data communications peripheral. The MPAC-100 provides a single channel for communications, however, portions of the second channel can be utilized to support some special circumstances. The SCC can be configured to satisfy a wide variety of serial communications applications. Some of its protocol capabilities include:

SDLC/HDLC (Bit Synchronous) Communications

- Abort sequence generation and checking
- Automatic zero insertion and deletion
- Automatic flag insertion between messages
- Address field recognition
- I-field residue handling
- CRC generation and detection
- SDLC loop mode with EOP recognition / loop entry and exit

Byte-oriented Synchronous Communications

- Internal / external character synchronization
- 1 or 2 sync characters in separate registers
- Automatic Cyclic Redundancy Check (CRC) generation / detection

Asynchronous Communications

- 5, 6, 7, or 8 bits per character
- 1, 1-1 / 2, or 2 stop bits
- Odd, even, or no parity
- Times 1, 16, 32, or 64 x clock modes
- Break generation and detection
- Parity, overrun and framing error detection

NRZ, NRZI, or FM encoding/decoding

8.1 Accessing the registers

The mode of communication desired is established and monitored through the bit values of the internal read and write registers. The register set of the SCC includes 16 write registers and 9 read registers. These registers only occupy four address locations, which start at the MPAC-100's physical base address that is configured via the on board switches. This and all other addresses are referenced from this base address in the form Base+Offset. An example of this is Base+1 for the SCC Control Port, Channel A.

There are two register locations per SCC channel, a data port and a control port. Accessing the internal SCC registers is a two step process that requires loading a register pointer to perform the addressing to the correct data register. The first step is to write to the control port the operation and address for the appropriate channel. The second step is to either read data from or write data to the control port. The only exception to this rule is when accessing the transmit and receive data buffers. These registers can be accessed with the two step process described or with a single read or write to the data port. The following examples illustrate how to access the internal registers of the SCC. Table 3 on page 26 describes the read registers and Table 4 on page 27 describes the write registers for each channel.

The MPAC-100 has been designed to assure that all back to back access timing requirements of the SCC are met without the need for any software timing control. The standard of adding jmp $+2 between I/O port accesses is not required when accessing the MPAC-100.

Example 1: Enabling the transmitter on channel A.

```
mov   dx, base    ; load base address
add   dx, ContA   ; add control reg A offset (1)
mov   al, 05H     ; write the register number
out   dx, al
mov   al, 08H     ; write the data to the register
out   dx, al
```

Example 2: Monitoring the status of the transmit and receive buffers in RR0 of Channel A. Register 0 is addressed by default if no register number is written to WR0 first.

```
mov   dx, base    ; load base address
add   dx, ContA   ; add control reg A offset (1)
in    ax, dx      ; read the status
```

Example 3: Write data into the transmit buffer of channel A.

```
mov   dx, base    ; load base address
out   dx, al      ; write data in ax to buffer
```

Example 4: Read data from the receive buffer of channel A.

```
mov   dx, base    ; load base address
in    al, dx      ; write data in ax to buffer
```

RR0	Transmit, Receive buffer statuses and external status
RR1	Special Receive Condition status, residue codes, error conditions
RR2	Modified Channel B interrupt vector and Unmodified Channel A interrupt vector
RR3	Interrupt Pending bits
RR6	LSB of frame byte count register
RR7	MSB of frame byte count and FIFO status register
RR8	Receive buffer
RR10	Miscellaneous status parameters
RR12	Lower byte of baud rate time constant
RR13	Upper byte of baud rate time constant
RR15	External/Status interrupt information

Table 3 --- SCC read register description

The SCC can perform three basic forms of I/O operations: polling, interrupts, and block transfer. Polling transfers data, without interrupts, by reading the status of RR0 and then reading or writing data to the SCC buffers via CPU port accesses. Interrupts on the SCC can be sourced from the receiver, the transmitter, or External/Status conditions. At the event of an interrupt, Status can be determined, then data can be written to or read from the SCC via CPU port accesses. Further information on this subject is found on page 23. For block transfer mode, DMA transfers are used, so this type of operation is not supported on the MPAC-100.

The SCC incorporates additional circuitry supporting serial communications. This circuitry includes clocking options, baud rate generator (BRG), data encoding, and internal loopback. The SCC may be programmed to select one of several sources to provide the transmit and receive clocks. These clocks can be programmed in WR11 to come from the RTxC pin, the TRxC pin, the output of the BRG, or the transmit output of the DPLL. The MPAC-100 uses the TRxC pin for its clock-on-transmit and the RTxC pin

for its clock-on-receive. Programming of the clocks should be done before enabling the receiver, transmitter, BRG, or DPLL.

WR0	Command Register, Register Pointer, CRC initialization, and resets for various modes
WR1	Interrupt control, Wait / DMA request control
WR2	Interrupt vector
WR3	Receiver initialization and control
WR4	Transmit / Receive miscellaneous parameters and codes, clock rate, stop bits, parity
WR5	Transmitter initialization and control
WR6	Sync character (1st byte) or SDLC address field
WR7	Sync character (2nd byte) or SDLC Flag
WR7'	Special HDLC Enhancement Register
WR8	Transmit buffer
WR9	Master interrupt control and reset
WR10	Miscellaneous transmitter / receiver control bits, NRZI, NRZ, FM coding, CRC reset
WR11	Clock mode and source control
WR12	Lower byte of baud rate time constant
WR13	Lower byte of baud rate time constant
WR14	Miscellaneous control bits: baud rate generator, DPLL control, auto echo
WR15	External / Status interrupt control

Table 4 --- SCC write register description

For complete information regarding the SCC registers please refer to Zilog's Z85230 technical manual.

8.2 Baud Rate Generator Programming

The baud rate generator (hereafter referred to as the BRG) of the SCC consists of a 16-bit down counter, two 8-bit time constant registers, and an output divide-by-two. The time constant for the BRG is programmed into WR12 (least significant byte) and WR13 (most significant byte). The equation relating the baud rate to the time constant is given below while Table 5 shows the time constants associated with a number of popular baud rates when using the standard MPAC-100 9.8304 MHz clock.

$$Time\_Const = Clock\_Frequency / 2 * Baud\_Rate * Clock\_Mode$$

Where: Clock_Frequency = 9.8304×10^6
Clock_Mode = 1, 16, 32, or 64
Baud_Rate = desired baud rate

Baud Rate	Time Constant	
38400	126	007E (hex)
19200	254	00FE (hex)
9600	510	01FE (hex)
4800	1022	03FE (hex)
2400	2046	07FE (hex)
1200	4094	0FFE (hex)
600	8190	1FFE (hex)
300	16382	3FFE (hex)

(for Clock_Frequency = 9.8304 MHz)

Table 5 --- time constants for common baud rates

8.3 SCC Data Encoding Methods

The SCC provides four different data encoding methods, selected by bits 6 and 5 in WR10. These four include NRZ, NRZI, FM1 and FM0. The SCC also features a digital phase-locked loop (DPLL) that can be programmed to operate in NRZI or FM modes. Also, the SCC contains two features for diagnostic purposes, controlled by bits in WR14. They are local loopback and auto echo.

For further information on these subjects or any others involving the SCC contact Zilog for a complete technical manual.

8.4 Support for SCC Channel B

The MPAC-100 is a single-channel device. Portions of SCC channel B are used to augment channel A. Channel B cannot be used for transmit, but may be used for receive, subject to certain limitations.

8.4.1 Receive data and clock signals

The receive data signals RXDA and RXDB are tied together. The receive clock input signals RTxCA and RTxCB are also tied together. This can be useful in unusual applications. It would be possible to run the receiver and transmitter at different baud rates, using channel B's baud rate generator and receiver for the received data. Of course, the channel A transmitter and receiver can be run at different speeds simply by having external data clocks supplied to TRxCA and RTxCA from the cable.

The W/REQB signal is used to generate DMA requests between the SCC and the internal FIFOs if channel B is used for receive.

8.4.2 Extra clock support for channel A

The TRxCB clock output can be routed back to RTxCA as another way to use the channel B baud rate generator to derive an independent clock for the channel A receiver. This is controlled by the RCKEN bit in the Communications Register (see page 39).

8.4.3 Extra handshaking for channel A

The SCC does not provide a DSR input for either channel. The MPAC-100 routes the DSR signal from the connector to the DCDB input of the SCC. Software can therefore use DCDB as a surrogate for DSR on channel A.

8.4.4 Other signals are not used

All channel B signals not listed above are not available at the connector. The CTSB and SYNCB inputs are tied to their inactive states. The TXDB, DTR/REQB, and RTSB outputs are left open.

8.5 SCC Incompatibility Warnings

Due to the SCC implementation used by the MPAC-100, there are two minor incompatibilities that the software programmer must avoid.

8.5.1 Register Pointer Bits

In a Zilog 85230, the control port register pointer bits can be set in either channel. With the implementation on the MPAC-100, however, both parts of an SCC control port access must use the same I/O address.

IMPORTANT

The programmer must be certain not to mix channel usage during the two-part access of SCC control ports. It would be highly irregular for code to be written in

The following sequences will work:

Write Control Port A	(set pointer bits for desired register)
Read or Write Control Port A	(read or write desired channel A register)
Write Control Port B	(set pointer bits for desired register)
Read or Write Control Port B	(read or write desired channel B register)

The following sequences will NOT work:

Write Control Port A	(set pointer bits for desired register)
Read or Write Control Port B	(read or write desired channel B register)
Write Control Port B	(set pointer bits for desired register)
Read or Write Control Port A	(read or write desired channel A register)

8.5.2 Software Interrupt Acknowledge

The 85230's software interrupt acknowledge mechanism is not supported. Bit 5 of Write Register 9 (Software INTACK Enable) is forced to 0. Software must employ the "Interrupt Without Acknowledge" interrupt method using Read Registers 2 and 3 to process interrupts.

9 FIFO Operation

The MPAC-100 is equipped with 1024-byte internal FIFOs in the transmit and receive data paths. These FIFOs are implemented as extensions of the SCC's small internal FIFOs. They have been designed to be as transparent as possible to the software operating the MPAC-100. By using these FIFOs, it is possible to achieve high data rates despite the MPAC-100 not supporting DMA.

The FIFOs are disabled by default after card insertion, power-up, or a system reset.

9.1 Enabling and disabling the FIFOs

The FIFOs must be enabled or disabled as a pair. It is not possible to operate only the transmit FIFO or only the receive FIFO. The FIFOs are enabled by setting bit 2 of the Configuration Register to a logic 1. The FIFOs are disabled by clearing the same bit.

9.2 Accessing the FIFOs

When the FIFOs are enabled, they are accessed through either the channel A or channel B SCC Data Port address. Writing to Base+0 or Base+2 will cause a byte to be written into the transmit FIFO. Reading from Base+0 or Base+2 will cause a byte to be read from the receive FIFO.

The FIFOs cannot be accessed if they are disabled. If the FIFOs are disabled, reads or writes of the SCC Data Ports access the receive or transmit register of the appropriate SCC channel. Any control port writes of SCC write register 8 (transmit buffer) or control port reads of SCC read register 8 (receive buffer) directly access the SCC, whether the FIFOs are enabled or not.

9.2.1 Transmit FIFO

The transmit FIFO always services the transmitter of channel A of the SCC. If the FIFOs are enabled, an I/O write to either SCC Data Port (channel A or channel B) will write a byte to the transmit FIFO. If the FIFOs are not enabled, an I/O write to the SCC Data Port will instead write directly to the internal transmit buffer of the specified channel of the SCC.

9.2.2 Receive FIFO

The receive FIFO can service the receiver of either channel A or channel B of the SCC. If RXSRC (bit 1) of the Configuration Register (see page 41) is logic 1, the receive FIFO will service SCC channel B. If RXSRC is logic 0, the receive FIFO will service SCC channel A.

If the FIFOs are enabled, an I/O read from either SCC Data Port (channel A or channel B) will read a byte from the receive FIFO. If the FIFOs are not enabled, an I/O read from the SCC Data Port will instead read directly from the internal receive buffer of the specified channel of the SCC.

9.3 SCC configuration for FIFO operation

The interface between the SCC and the internal FIFOs uses the SCC's DMA request functions. The SCC must therefore be configured for DMA operation in order to use the internal FIFOs. In order to properly configure the SCC, certain bits in various SCC registers need to be set in a specific manner, as shown on the following pages.

Because the data transfer between the FIFOs and the SCC is controlled entirely by hardware, per-character transmit and receive interrupts should be disabled. Interrupts on transmit underruns and/or special receive conditions should usually be enabled so that end-of-frame conditions can be detected.

IMPORTANT

The DMA operation described in this section is between the SCC and the internal FIFOs, and is handled entirely by the MPAC-100 hardware.

DMA is not supported between the

The MPAC-100 is a single-channel device. Accordingly, most applications will use SCC channel A for both transmit and receive operations. It is possible, however, to use a limited portion of SCC channel B for receive operations (see page 29). The channel used for receive will determine how the SCC must be configured.

Do not enable the FIFOs until the SCC has been properly configured for DMA operation!

9.3.1 Using channel A for both transmit and receive

This is the mode in which most applications will run. Set RXSRC (bit 1) in the Configuration Register to logic 0. This will configure the MPAC-100 to use W/REQA for receive DMA and DTR/REQA for transmit DMA. In addition to any other desired SCC configuration, ensure that the following bits are set according to Table 6:

Register	Bit(s)	Value	Function
WR1A	7	1	Enable DMA request on W/REQA. This bit should be set after the other bits in WR1 are set as desired.
	6	1	Set W/REQA for DMA Request mode.
	5	1	Use W/REQA for receive.
	4-3	11 or 00	Enable receive interrupts on special conditions only (recommended), or disable them completely.
	1	0	Disable transmit interrupts.
WR14A	2	1	Enable DMA request-on-transmit on DTR/REQA.
WR15A	0	1	Enable WR7A'.
WR7A'	5	0	Assert transmit DMA request when entry location of internal FIFO is empty.
	4	1	Set DTR/REQA for W/REQA timing.

Table 6 --- Configuring the SCC for FIFO use with channel A only

9.3.2 Using channel B for receive

The MPAC-100 supplies only limited support for SCC channel B. This mode, therefore, is not recommended for most applications. Set RXSRC (bit 1) in the Configuration Register to logic 1. This will configure the MPAC-100 to use W/REQA for transmit DMA and W/REQB for receive DMA. In addition to any other desired SCC configuration, ensure that the following bits are set according to Table 7:

Register	Bit(s)	Value	Function
WR1A	7	1	Enable DMA request on W/REQA. This bit should be set after the other bits in WR1 are set as desired.
	6	1	Set W/REQA for DMA Request mode.
	5	0	Use W/REQA for transmit.
	1	0	Disable transmit interrupts.
WR14A	2	0	Disable DMA request-on-transmit on DTR/REQA.
WR15A	0	1	Enable WR7A'.
WR7A'	5	0	Assert transmit DMA request when entry location of internal FIFO is empty.
WR1B	7	1	Enable DMA request on W/REQB. This bit should be set after the other bits in WR1 are set as desired.
	6	1	Set W/REQB for DMA Request mode.
	5	1	Use W/REQB for receive.
	4-3	11 or 00	Enable receive interrupts on special conditions only (recommended), or disable them completely.

Table 7 --- Configuring the SCC for Rx DMA on channel B

9.4 FIFO status and control

Several registers are used to control the FIFOs and monitor their status. These registers are detailed in other chapters of this manual.

9.4.1 Interrupt status

Three interrupt statuses, listed in Table 8, can be generated by four events related to FIFO activity. In each case, a latched bit in the Interrupt Status Register is set to a logic 1 (see page 43). These bits are write-clear, meaning that software must write a 1 to a bit in order to clear it.

IMPORTANT

FIFO-related interrupts will occur only when the MPAC-100 interrupt source is set to INTSCC. See Table 10 on page 41 for details.

Event	Interrupt Status Register Bit	Comment
Transmit FIFO drained past the half-full mark	TX_FIFO (bit 1)	Software can write at least 512 bytes to the transmit FIFO.
Receive FIFO filled past the half-full mark	RX_FIFO (bit 2)	Software can read at least 512 bytes from the receive FIFO.
Receive data timeout with non-empty FIFO		Software can read bytes from the receive FIFO until the FIFO is empty.
Special receive pattern detected	RX_PAT (bit 3)	Software can read data from the receive FIFO as desired.

Table 8 --- FIFO-related interrupt statuses

IMPORTANT

Software can differentiate between the two types of RX_FIFO interrupts by examining the RXH bit in the FIFO Status Register. If RXH is clear (logic 0), the interrupt occurred because of a timeout.

9.4.2 Resetting the FIFOs

The FIFOs are automatically disabled and reset at powerup or when the MPAC-100 is inserted into a PCI socket. The transmit and receive FIFOs can also be independently reset by setting and clearing the appropriate bits in the FIFO Control Register. Resetting a FIFO sets the appropriate FIFO empty status bit and resets the FIFO's internal read and write pointers. The SCC's internal FIFOs are not affected when the internal FIFOs are reset.

The internal FIFOs cannot be reset while they are enabled! FIFO reset commands will be ignored if the internal FIFOs are enabled.

9.4.3 Reading current FIFO status

The FIFO Status Register is a read-only register which always indicates the current status of both the transmit and receive internal FIFOs. Each FIFO can be checked for empty, full, and half-full (or more) status at any time. For details, see Table 12 on page 44.

9.4.4 Controlling the FIFOs

The FIFO Control Register is a read-write register which can be used to reset either or both the receive and transmit internal FIFOs. Receive pattern detection and receive FIFO timeout modes are also controlled with this register. For details, see Table 13 on page 45.

9.5 Accessing the SCC while FIFOs are enabled

The SCC channel A and channel B control port registers are always accessible regardless of whether the internal FIFOs are enabled or disabled. While the FIFOs are enabled, SCC data port accesses are redirected to the FIFOs. Access to the SCC's transmit or receive registers while the FIFOs are enabled is possible indirectly by using the control port and register 8. Any writes of SCC Write Register 8 (transmit buffer) or reads of SCC Read Register 8 (receive buffer) will bypass the internal FIFOs.

9.6 Receive pattern detection

The internal FIFOs are most useful in bit-synchronous operational modes because the SCC can generate a Special Condition interrupt when the closing flag of a bit-synchronous frame is received. This allows the SCC to run with per-character receive interrupts disabled while DMA transfers occur between the SCC and internal FIFOs.

Byte-synchronous modes such as bisync, however, do not benefit from such a hardware assist for detecting the end-of-frame condition. On the contrary, with byte-oriented protocols it is usually necessary to check each byte received against a table of special function codes (e.g. SYNC, PAD, SDI, STX, EDI, ETX, etc.) to determine where data and frames begin and end. Unless the frames are of a fixed length, it is therefore difficult to use DMA with byte-synchronous modes. This would seem to preclude the use of the MPAC-100's internal FIFOs with byte-oriented protocols.

To make the internal FIFOs more useful in byte-synchronous modes, the MPAC-100 can watch for a given character to be transferred consecutively a specific number of times from the SCC into the receive FIFO. When this occurs, the RX_PAT bit in the Interrupt Status Register (see page 43) is set. For instance, the MPAC-100 can watch for the end-of-text character to be received, or for three consecutive pad characters to be received.

For byte-synchronous operation with simple unique markers in the data stream, this feature may be quite useful. Even if it is not, however, the MPAC-100 can certainly be operated with per-character interrupts enabled and the internal FIFOs disabled. The tradeoff will be a heavier interrupt burden and possibly somewhat lower throughput.

NOTE

While most useful in byte-synchronous modes, the receive pattern detection feature can be used in any operational mode.

9.7 Receive FIFO timeout

With asynchronous operational modes, the same problem exists. Namely, how is one to determine when a reception is complete? While the receive pattern detection may be useful here, the MPAC-100 also offers a timeout feature on the internal receive FIFO.

If the internal FIFO is not empty and a time interval equal to a specified number of character-times has elapsed without any further data being received, a receive FIFO interrupt is generated and RX_FIFO bit in the Interrupt Status Register (see page 43) is set. A character-time is approximated by counting eight ticks of the bit clock.

To use this feature, the receive clock must be output on TRxCA. It can come from either an internal source or from the channel A baud rate generator. While the RTxCA signal is typically used for a receive clock, it is not capable of being an output, so the TRxCA signal must be used instead. Depending on the application, this may force the transmit and receive clocks to be the same. For most asynchronous applications, this should not pose a problem.

10 Communications Register

The Communications Register is used to set options pertaining to the clocks. The source and type of clock to be transmitted or received can be specified. External synchronization and RS-232 DTE test modes and can also be controlled with this register. The address of the Communications Register is Base+4. Table 9 details its bit definitions.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM ST	EXTSYNC	LLEN	RLEN or SW_SYNC	RCKEN	TCKEN	0	0

Table 9 --- Communications Register - Read / Write

Bit 7: TM ST --- Test Mode Status:
This bit can be used to read the status of the Test Mode signal on a DTE, allowing the user to monitor the signal without generating any interrupts.

Bit 6: EXTSYNC --- External Sync Enable:
If this bit is set (logic 1), software-controlled sync is disabled and the SCC's SYNCA input is driven by the signal coming on pin 10 of the DB-25 connector.

Bit 5: LLEN --- Local Loopback Enable:
When set (logic 1), this bit allows the DTE to test the functioning of the DTE / DCE interface and the transmit and receive sections of the local DCE. The DCE device must support local loopback for this to work. When cleared (logic 0), no testing occurs. LLEN can also be used as a software-controlled general-purpose output.

Bit 4: RLEN --- Remote Loopback Enable: SW_SYNC --- Software Sync On:
When the 'C' option is used with the client driver or enabler, this bit functions as SW_SYNC, otherwise it functions as RLEN.

RLEN ('C' option not used)
If this bit is set (logic 1), the DTE can test the transmission path through the remote DCE to the remote DTE interface and the return transmission path. The remote device must support remote loopback for this to work. When cleared (logic 0), no testing occurs.

SW_SYNC ('C' option is used)
This bit is used to drive the active-low SYNC input of the channel A receiver. The SYNC signal is asserted when this bit is set (logic 1), and is deasserted when this bit is clear (logic 0). This is useful in situations where it is necessary to receive unformatted serial data, as it allows the SCC receiver to be manually placed into sync under program control. This bit is ignored if bit 6 is set (logic 1).

Bit 3: **RCKEN --- Receive Clock Source:**
When set (logic 1), this bit allows the receive clock (RCLK) signal to be generated by the TRxC pin on channel B of the SCC. When cleared (logic 0), RCLK is received on pin 17 of the DB-25 connector. In either case, RCLK is always transmitted on pin 11 of the DB-25 connector.

Bit 2: **TCKEN --- Transmit Clock Source:**
When set (logic 1), this bit allows the transmit clock (TCLK) to be generated by the TRxC pin on channel A of the SCC. When cleared (logic 0), the DTE receives TCLK on pin 15 of the DB-25 connector. In either case, TCLK is always transmitted on pin 24 of the DB-25 connector.

Bits 1-0: **Reserved, always 0.**

IMPORTANT

Local Loopback and Remote Loopback cannot be enabled simultaneously. Bits 5 and 4 of the Communications Register should therefore not be set (logic 1) simultaneously.

11 Configuration Register

The Configuration Register is used to set the interrupt source and enable the interface between the SCC and the internal FIFOs. The address of this register is Base+5. Table 10 details the bit definitions of the register.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	0	INTS1	INTS0	0	FIFOEN	RXSRC	0

Table 10 --- Configuration Register - Read / Write

Bit 7: **Internal Data FIFOs Present --- Reserved, always 1.** This bit can be used as an indicator that internal data FIFOs are present. Other MPA-series products that are not equipped with internal or external data FIFOs will return 0 in this bit location.

Bit 6: **Reserved, always 0.**

Bits 5-4: **INTS1, INTS0 --- Interrupt Source and Enable Bits:** These two bits determine the source of the interrupt. The two sources are interrupt from the SCC (INTSCC), and interrupt on Test Mode (INTTM). Only one interrupt source can be active at a time. Below is the mapping for these bits. Note that FIFO-related interrupts will occur only when INTSCC is chosen.

INTS1	INTS0	Interrupt Source
0	0	Interrupts disabled
0	0	reserved
1	0	INTSCC
1	1	INTTM

Bit 3: **Reserved, always 0.**

Bit 2: **FIFOEN --- Internal data FIFO enable:** If this bit is set (logic 1), the internal data FIFOs are enabled. If this bit is clear (logic 0), the internal data FIFOs are disabled. (See page 31 for full details on FIFO use.)

Bit 1: **RXSRC --- Receive FIFO DMA Source:** This bit determines which SCC pins are used to control transmit and receive DMA transactions between the SCC and the internal FIFOs (when enabled). The transmit data FIFO is always used with SCC channel A. The receive data FIFO may be used with SCC channel A by setting RXSRC to logic 0, or with SCC channel B by setting RXSRC to logic 1. (See page 29 for information on using channel B.)

	RXSRC = 0	RXSRC = 1
Receive DMA	W / REQA	W / REQB
Transmit DMA	DTR / REQA	W / REQA

Bit 0: **Reserved, always 0.**

12 Interrupt Status Register

The Interrupt Status Register is used to determine the cause of an interrupt generated by the MPAC-100. The address of this register is Base+8. Table 11 details the bit definitions of the register. **The interrupt source in the Configuration Register (see page 41) must be set to INTSCC for any of the statuses indicated by this register to occur.** This register can be ignored if the internal FIFOs are not being used.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	RX_PAT	RX_FIFO	TX_FIFO	SCC

Table 11 --- Interrupt Status Register - Read Only / Write Clear

Bits 7-4: Reserved, always 0.

Bit 3: RX_PAT --- Receive Pattern Interrupt: The receive pattern interrupt occurs when the character set in the Receive Pattern Character Register is detected 'n' consecutive times in the received data stream, where 'n' is the value set in the Receive Pattern Count Register. This bit is set (logic 1) to indicate the interrupt. It remains set until cleared by writing a '1' to this bit.

Bit 2: RX_FIFO --- Receive FIFO Interrupt:
The receive FIFO interrupt occurs when the number of bytes held in the internal receive FIFO rises above the half-full mark, or when a receive FIFO timeout occurs. This bit is set (logic 1) to indicate the interrupt. It remains set until cleared by writing a '1' to this bit.

Bit 1: TX_FIFO --- Transmit FIFO Interrupt: The transmit FIFO interrupt occurs when the number of bytes held in the internal transmit FIFO falls below the half-full mark. This bit is set (logic 1) to indicate the interrupt. It remains set until cleared by writing a '1' to this bit.

Bit 0: SCC --- SCC Interrupt:
If this bit is set (logic 1), the SCC has generated an interrupt. Software should clear the interrupt condition by performing appropriate service on the SCC. This bit is not latched.

13 FIFO Status Register

The FIFO Status Register is used to return current status information about the internal FIFOs. The address of this read-only register is Base+9. Table 12 details the bit definitions of the register. This register can be ignored if the internal FIFOs are not being used.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	RXF	RXH	RXE	0	TXF	TXH	TXE

Table 12 --- FIFO Status Register - Read Only

Bit 7: **Reserved, always 0.**

Bit 6: **RXF --- Receive FIFO Full:** This bit is set (logic 1) when the internal receive FIFO is completely full. The FIFO will accept no more data from the SCC.

Bit 5: **RXH --- Receive FIFO Half Full:**
This bit is set (logic 1) while the internal receive FIFO is at least half-full.

Bit 4: **RXE --- Receive FIFO Empty**:
This bit is set (logic 1) when the internal receive FIFO is completely empty.

Bit 3: **Reserved, always 0.**

Bit 2: **TXF --- Transmit FIFO Full:**
This bit is set (logic 1) when the internal transmit FIFO is completely full. Further writes to the internal transmit FIFO will be ignored.

Bit 1: **TXH --- Transmit FIFO Half Full:**
This bit is set (logic 1) while the internal transmit FIFO is at least half-full.

Bit 0: **TXE --- Transmit FIFO Empty**:
This bit is set (logic 1) when the internal transmit FIFO is completely empty.

14 FIFO Control Register

The FIFO Control Register is used to control the internal data FIFOs. The address of this register is Base+A (hex). Table 13 details the bit definitions of the register. This register can be ignored if the internal FIFOs are not being used.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	EN_PAT	EN_TO	RX_RESET	0	0	0	TX_RESET

Table 13 --- FIFO Control Register - Read / Write

Bit 7: **Reserved, always 0.**

Bit 6: **EN_PAT --- Enable Receive Pattern Detection:** Set this bit (logic 1), to enable the receive pattern detection circuitry. Clear this bit (logic 0), to disable pattern detection. See page 37 for details on the receive pattern detection feature.

Bit 5: **EN_TO --- Enable Receive Timeout:**
Set this bit (logic 1), to enable the internal receive FIFO timeout. Clear this bit (logic 0), to disable the receive FIFO timeout. See page 38 for details on the receive FIFO timeout feature.

Bit 4: **RX_RESET --- Reset Receive FIFO**:
Set (logic 1), then clear (logic 0) this bit to reset the internal receive FIFO. The FIFO can be reset only when it is disabled.

Bits 3-1: **Reserved, always 0.**

Bit 0: **TX_RESET --- Reset Transmit FIFO**:
Set (logic 1), then clear (logic 0) this bit to reset the internal transmit FIFO. The FIFO can be reset only when it is disabled.

15 Receive Pattern Character Register

The Receive Pattern Character Register is used to set the character value to be used in receive pattern detection. The address of this register is Base+B (hex). This register can be ignored if the internal FIFOs are not being used.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
character value (0-255)							

Table 14 --- Receive Pattern Character Register - Read / Write

Bits 7-0: **Receive Pattern Character:** This is the numeric value of the character to be detected. See page 37 for details on the receive character pattern detection feature.

16 Receive Pattern Count Register

The Receive Pattern Count Register is used to set the counter value to be used in receive pattern detection. The address of this register is Base+C (hex). This register can be ignored if the internal FIFOs are not being used.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
counter value (0-255)							

Table 15 --- Receive Pattern Count Register - Read / Write

Bits 7-0: **Receive Pattern Count:**
This value is the number of times that the character stored in the Receive Pattern Character Register (see page 46) must be consecutively detected for the receive character pattern detect interrupt to be generated. See page 37 for details on the receive character pattern detection feature.

17 Receive FIFO Timeout Register

The Receive FIFO Timeout Register is used to control the operation of the internal receive FIFO timeout feature. The address of this register is Base+D (hex). This register can be ignored if the internal FIFOs are not being used. See page 38 for details on the receive FIFO timeout feature.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
X16_MODE	0	timeout interval (0-63)					

Table 16 --- Receive FIFO Timeout Register - Read / Write

Bit 7: **X16_MODE --- Clock Mode:**
If this bit is set (logic 1), the data clock is divided by 16 (prescaled) before it is fed to the timeout circuitry. This is useful for asynchronous operation. If this bit is clear (logic 0), the data clock is not prescaled.

Bit 6: **Reserved, always 0.**

Bits 5-0: **Timeout Interval:**
This is the number of character-times that must elapse before a non-empty internal receive FIFO will trigger a timeout condition. This interval assumes eight bits per character, so it will be an approximation for modes running at settings other than eight bits per character.

18 External Connections

The MPAC-100 is configured as a Data Terminal Equipment (DTE) device, meeting the RS-232-D standard using a DB-25 male connector. There is no DCE version available.

The control signals the DTE can generate are Request To Send (RTS) and Data Terminal Ready (DTR). It can receive the signals Carrier Detect (DCD), Clear to Send (CTS), and Data Set Ready (DSR). All the control signals are controlled through channel A of the SCC, with the exception of the DSR signal, which is received on the DCDB pin on channel B. (The SCC has no actual DSR inputs.)

The DTE can transmit its transmit clock (TCLK) from the TRxCA pin of the SCC, or receive TCLK on the same pin. The DTE can also receive its receive clock (RCLK) on the RTxC pins on channels A & B of the SCC, or can generate RCLK using the TRxCB pin. TCLK and RCLK can also be internally sourced from the channel A baud rate generator.

Figure 1 shows the DTE clock configuration. On the left are the SCC clock pins and the clock enable bits from the Communications Register. On the right are the signals at the DB-25 connector. Figure 2 illustrates the connector pinout.

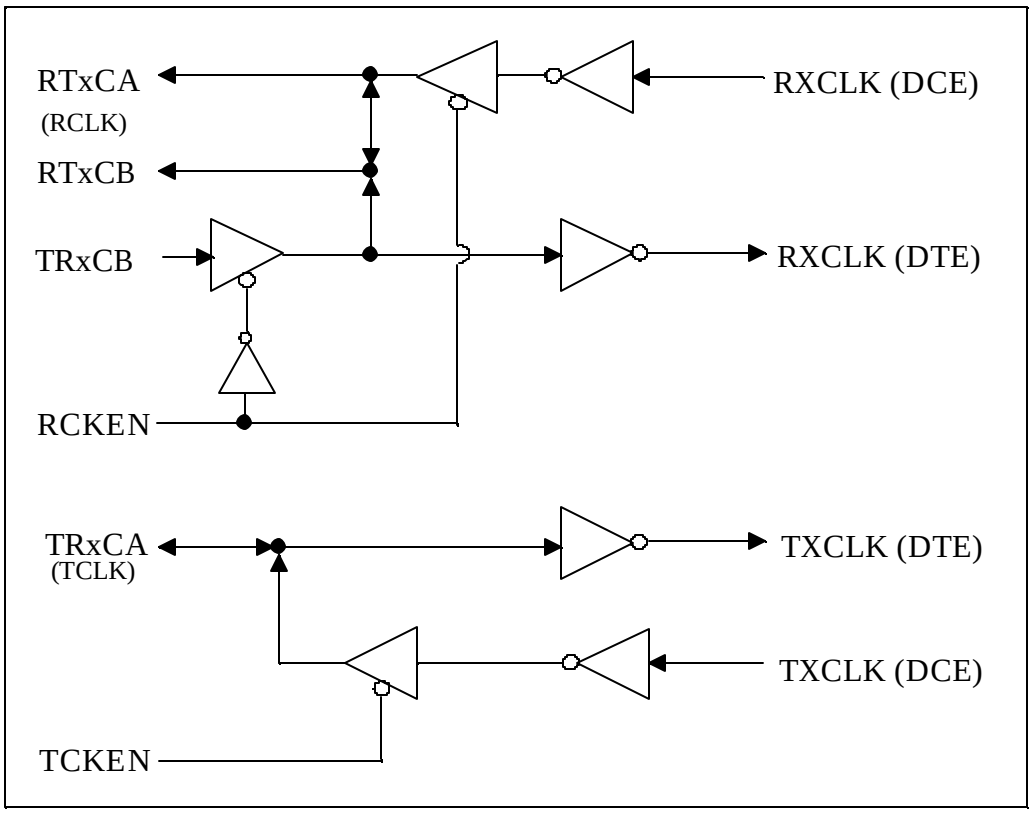


Figure 1 --- MPAC-100 Clock Configuration

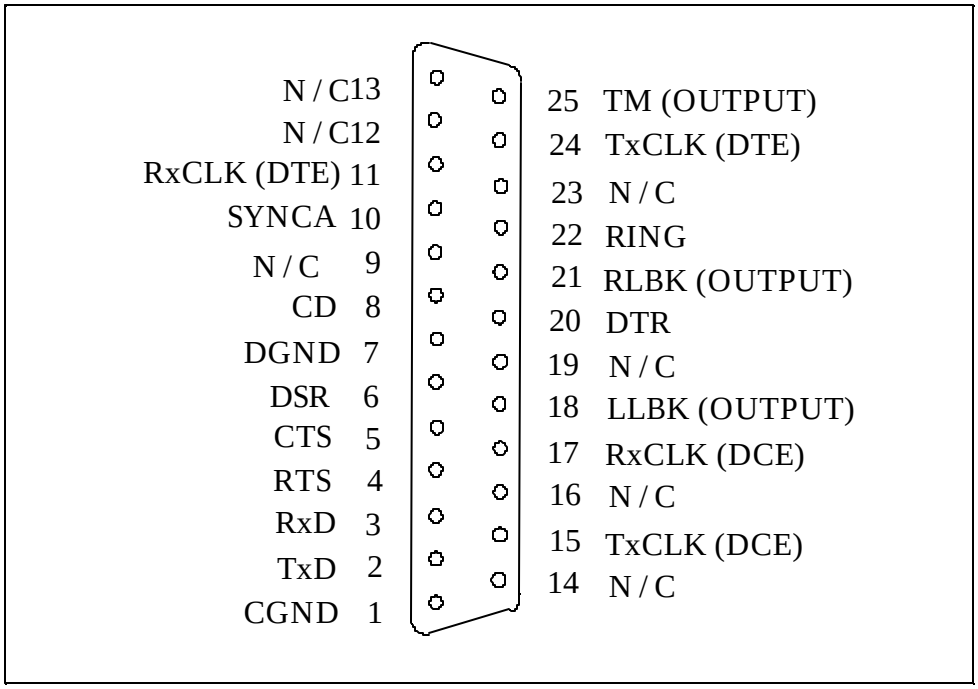


Figure 2 --- MPAC-100 Output Connector

The testing signals the DTE can generate are Local Loopback (LL) and Remote Loopback (RL). These signals are asserted with certain bits in the Communications Register. When a Test Mode (TM) condition is received from the DCE, an interrupt can optionally be generated.

18.1 5V fuse (pin 9)

Pin 9 will have a 5volt fuse tied to VDD on the other end. This is compatible with the MPAP-100 series cards.

18.2 SYNCA (pin 10)

If EXTSYNC (bit 6) in the Communications Register is set to a logic 1, the SYNCA signal from the connector is used to drive the active-low SYNC input of SCC channel A. The signal is inverted by the RS-232 receiver, so a positive voltage on pin 10 will assert SYNCA. The SCC must be specifically programmed to recognize external synchronization.

18.3 RING (pin 22)

If Card and Socket Services has set the SIGCHG bit in the PCI Configuration Status Register to a logic 1, the RING signal is routed to the STSCHG line on the PCI bus. The signal is inverted by the RS-232 receiver, so a positive voltage on pin 22 will assert STSCHG.

Table 17 shows the pin configuration of the MPAC-100 DTE connector. The definitions of the interchange circuits according to the RS-232-D standard can be found starting on page 52.

Pin	To DTE	From DTE	Signal	RS-232-D Circuit	SCC Pin or Register Bit
1			CGND		
2		X	TXD	BA	TxDA pin
3	X		RXD	BB	RxDA pin
4		X	RTS	CA	RTSA pin
5	X		CTS	CB	CTSA pin
6	X		DSR	CC	DCDB pin
7			DGND	AB	
8	X		CD	CF	DCDA pin
9			N / C		
10	X		SYNCA	*	SYNCA pin
11		X	RXCLK (DTE)	*	RTxCA or TRxCB pin
12			N / C		
13			N / C		
14			N / C		
15	X		TXCLK (DCE)	DB	TRxCA pin
16			N / C		
17	X		RXCLK (DCE)	DD	RTxCA pin
18		X	LLBK	LL	Comm. Reg. bit 5
19			N / C		
20		X	DTR	CD	DTR/REQA pin
21		X	RLBK	RL	Comm. Reg. bit 4
22	X		RING	CE	PCI STSCHG signal
23			N / C		
24		X	TXCLK (DTE)	DA	TRxCA pin
25	X		TEST MODE	TM	Comm. Reg. bit 7
* Not included in the official RS-232-D specification					

Table 17 --- Connector Pin Definitions

18.4 Null-modem cables

The MPAC-100 does not use a standard asynchronous PC serial port connector pinout. Typical off-the-shelf null-modem cables cannot be used with this card!

19 DTE Interface Signals

CIRCUIT AB - SIGNAL GROUND

- CONNECTOR NOTATION: DGND
- DIRECTION: Not applicable

This conductor directly connects the DTE circuit ground to the DCE circuit ground.

CIRCUIT BA - TRANSMITTED DATA

- CONNECTOR NOTATION: TXD
- DIRECTION: To DCE

This signal transfers the data generated by the DTE through the communication channel to one or more remote DCE data stations.

CIRCUIT BB - RECEIVED DATA

- CONNECTOR NOTATION: RXD
- DIRECTION: From DCE

This signal transfers the data generated by the DCE, in response to data channel line signals received from a remote DTE data station, to the DTE.

CIRCUIT CA - REQUEST TO SEND

- CONNECTOR NOTATION: RTS
- DIRECTION: To DCE

This signal controls the data channel transmit function of the local DCE and, on a half-duplex channel, the direction of the data transmission of the local DCE.

CIRCUIT CB - CLEAR TO SEND

- CONNECTOR NOTATION: CTS
- DIRECTION: From DCE

This signal indicates to the DTE whether the DCE is conditioned to transmit data on the communication channel.

CIRCUIT CC - DCE READY (DATA SET READY)

- CONNECTOR NOTATION: DSR
- DIRECTION: From DCE

This signal indicates the status of the local DCE by reporting to the DTE device that a communication channel has been established.

CIRCUIT CD - DTE READY (DATA TERMINAL READY)

- CONNECTOR NOTATION: DTR
- DIRECTION: To DCE

This signal controls the switching of the DCE to the communication channel. The DTE will generate this signal to prepare the DCE to be connected to or removed from the communication channel.

CIRCUIT CE - RING INDICATOR

- CONNECTOR NOTATION: RING
- DIRECTION: From DCE

This signal indicates that a ringing signal is being received on the communication channel.

CIRCUIT CF - RECEIVED LINE SIGNAL DETECT (CARRIER DETECT)

- CONNECTOR NOTATION: CD
- DIRECTION: From DCE

This signal indicates to the DTE whether the DCE is conditioned to receive data from the communication channel, but does not indicate the relative quality of the data signals being received.

CIRCUIT DA - TRANSMIT SIGNAL ELEMENT TIMING (DTE SOURCE)

- CONNECTOR NOTATION: TXCLK (DTE)
- DIRECTION: To DCE

This signal, generated by the DTE, provides the DCE with element timing information pertaining to the data transmitted by the DTE. The DCE can use this information for its received data.

CIRCUIT DB - TRANSMIT SIGNAL ELEMENT TIMING (DCE SOURCE)

- CONNECTOR NOTATION:TXCLK (DCE)
- DIRECTION: From DCE

This signal, generated by the DCE, provides the DTE with element timing information pertaining to the data transmitted to the DCE. The DCE can use this information for its received data.

CIRCUIT DD - RECEIVER SIGNAL ELEMENT TIMING (DCE SOURCE)

- CONNECTOR NOTATION: RXCLK (DCE)
- DIRECTION: From DCE

This signal, generated by the DCE, provides the DTE with element timing information pertaining to the data transmitted by the DCE. The DTE can use this information for its received data.

CIRCUIT LL - LOCAL LOOPBACK

- CONNECTOR NOTATION: LLBK
- DIRECTION: To DCE

This signal provides a means whereby a DTE may check the functioning of the DTE/DCE interface and the transmit and receive sections of the local DCE.

CIRCUIT RL - REMOTE LOOPBACK

- CONNECTOR NOTATION: RLBK
- DIRECTION: To DCE

This signal provides a means whereby a DTE or a facility test center may check the transmission path up to and through the remote DCE to the DTE interface and the similar return transmission path.

CIRCUIT TM - TEST MODE

- CONNECTOR NOTATION: TEST MODE
- DIRECTION: From DCE

This signal indicates to the DTE that the DCE is in a test condition. The DCE generates this signal when it has received a local loopback or remote loopback signal from the DTE.

20 PCI Resource Map

Listed below are the PCI resources used by the MPAC-100. Such information may be of use to customers writing their own device drivers or other custom software.

(all numbers in hex)

PCI Vendor ID:	0x135C	Quatech, Inc.
PCI Device ID:	0x00F0	MPAC-100
PCI Class Code		
Base class:	0xFF	Undefined Class
Subclass:	0x00	
Interface:	0x00	
IRQ sourced by:	INTA#	
Base address 0:	0x80 bytes I/O	Reserved region
Base address 1:	0x10 bytes I/O	MPAC-100 I/O Address Space

21 Specifications

Bus interface:	PCI, 32-bit bus, 5 volt only AMCC 5920 PCI Controller
Physical Dimensions:	approx. 4.5” x 2.5”
Controller:	Zilog Z85230 20-MHz Serial Communications Controller (SCC)
DTE Interface:	Male D-25 connector 5 volt fuse on pin 9
Transmit drivers:	SP211HB RS-232 compatible, 600 kbps typical maximum data rate
Receive buffers:	SP211HB RS-232 compatible, 600 kbps typical maximum data rate
I/O Address range:	Sixteen-byte contiguous range required, determined by PCI system
Interrupt levels:	One IRQ required, determined by PCI system
DMA channels:	Not supported by PCI bus
Power requirements:	115 mA at +5 volts, typical

MPAC-100 User's Manual
Revision 1.01
June 2001
P / N 940-0090-220